Department of Mechanical Engineering and Mechanics

Drexel University College of Engineering

Optimal Shipboard Power System Management via Mixed Integer Dynamic Programming

Harry G. Kwatny & Edoe Mensah
MEM Department
Drexel University
Philadelphia, PA 19104, USA
hkwatny@coe.drexel.edu, edmensah@drexel.edu

Dagmar Niebur
ECE Department
Drexel University
Philadelphia, PA 19104, USA
niebur@drexel.edu

Carole Teolis
Techno-Sciences, Inc.
10001 Derekwood Lane
Lanham, MD 20706, USA
carole@technosci.com

***Abstract*— Power systems involve continuous and discrete components and controls. The modeling of 'hybrid' power systems using a logical specification to define the transition dynamics of the discrete subsystem is described. A computational tool for reduction of the logical specification to a set of inequalities is discussed. The use of the transformed model in a dynamic programming approach to the design of optimal feedback controls is described. Examples are given.**

I. Introduction

Maintaining power flow to vital loads following component failure(s) is a central goal of power system management including electric shipboard distribution systems. While the dynamics of the final phase of power system failure is now well understood, e.g. [1], [2], [3], [4], the picture is not complete because the system collapse is usually preceded by a period of discrete events associated with the action of various protection systems intended to prevent, or at least limit the scope, of any failure. It is an unfortunate fact that these systems frequently fail to achieve that goal - and worse, they sometimes amplify the effect of a small disturbance into a major outage. In this paper we seek to design a power management system that optimizes the discrete actions in order to insure continuity of service to the vital loads. We describe a modeling approach that captures both the discrete and continuous aspects of the power system and show how dynamic programming can be applied to derive optimal control strategies. New computational tools are summarized and examples are given.

The underlying issue is how to model, analyze and synthesize systems consisting of both complex nonlinear continuous dynamics and discrete event dynamics. A power system's continuous dynamics might include a classical ordinary differential equation (ODE) or differential-algebraic equation (DAE) model of the network with generators and loads and also continuous controllers like governors and automatic voltage regulators. Discrete event dynamics can be defined by a discrete finite automaton (DFA) [5] that models various discrete controllers like tap-changing transformers, capacitor banks, load shedding devices and protection systems [6]. Thus, the system can be modeled as a hybrid automaton [7]. While the hybrid automaton model is a convenient theoretical tool, other forms of models are far more convenient for control system design and some other computational purposes. Such models include the 'mixed logical dynamic system' (MLD) [8], [6] or a modified version that we call the 'mixed integer dynamic program' (MIDP).

The action of the DFA is most easily understood in terms of a transition diagram that describes how specific events cause transitions from one discrete state (or mode) to another. The transition diagram is ordinarily translated into a formal set of transition equations. In our approach, we model the transition diagram by a logical statement (or specification).

In this paper, we discuss a tool, implemented in Mathematica , that converts any logical specification into a set of mixed-integer formulas (IP formulas). Thus, the transition specification for the automaton is converted into a set of inequalities involving Boolean variables. Our work extends earlier work in this area reported in [9]. Many decision problems are most naturally formulated in terms of logical specifications, but are more easily solve by mathematical programming. Consequently, the idea of reducing logical specifications into IP formulas has along history, see for example [10].

The IP formulas are used in computing the optimal control strategy. Our approach derives a feedback policy based on finite horizon dynamic programming. We implement the resulting control policy either as a receding horizon or periodic controller as appropriate. To do the computations efficiently, we need to exploit the special structure of the power system decision problem.

In the dynamic programming approach, working backward in time, at each state it is necessary to carry out a minimization process involving continuous and integer (binary) variables to obtain the optimal control. A computational method has been implemented in *Mathematica*, which provides several tools for working with mixed variables. The problems of interest have considerable special structure that can be exploited. For example, we have many inequality constraints which implies that we should employ a constraint driven procedure. Moreover, most of the constraints are linear in binary variables. Accordingly, a specialized and novel optimization procedure was built around the *Mathematica* function `Reduce`.

The design of optimal controls for hybrid systems is currently a problem of great interest. At least three approaches to optimal control design have been considered: Bellman's

Principle of optimality and dynamic programming [11], the Pontryagin maximum principle [12], and mixed integer mathematical programming [8]. Each of these have advantages and disadvantages. The principle of optimality is quite general and applies directly to hybrid systems. In addition, unlike the other two approaches, dynamic programming leads directly to a feedback (or closed loop) controller as opposed to an open loop controller. These two considerations make dynamic programming a very compelling tool even though it suffers the 'curse of dimensionality'. This last fact represents a challenge for large systems that can sometimes be addressed by exploiting the special structure of specific problems. Applications of dynamic programming to hybrid systems include [13], [14], [11].

In Section II we provide a specific definition of the problems considered herein. Section III describes the main concern of this paper, namely the reduction of a logical specification for the discrete subsystem to a set of inequalities. The goal of this research is to use this formulation of the hybrid power system model to design optimal controllers. How this is accomplished is described in Section IV. Examples are given in Section V. The examples include a power conditioning system, a DC-DC boost converter, and a power management system. In each one of these examples we describe the model emphasizing the hybrid automaton and illustrating the conversion of the logical specification to IP formulas. In the case of the power conditioning system we also describe the setup and solution of the optimal control problem.

II. Problem Definition

A. Modeling

The class of hybrid systems to be considered is defined as follows. The system operates in one of m modes denoted $q_1, \ldots, q_m$. We refer to the set of modes $Q = \{q_1, \ldots, q_m\}$ as the discrete state space. The discrete time dynamical equation describing operation in mode q_i is

$$x_{k+1} = f_{q_i}(x_k, u_k), \quad i = 1, \ldots, m \tag{1}$$

where $x \in X \subseteq R^n$ is the system continuous state and $u \in U \subseteq R^m$ is the continuous control. Transitions can occur only between certain modes. The set of admissible transitions is $E \subseteq Q \times Q$. It is convenient to view the mode transition system as a graph with elements of the set Q being the nodes and the elements of E being the edges. We assume that transitions are instantaneous and take place at the beginning of a time interval. So, if a system transitions from mode q_1 to q_2 at time k we would write $q(k) = q_1, q(k^+) = q_2$. We do not consider 'impulsive' events [11]. In other words, the continuous state trajectories are continuous through the event, i.e., $x(k) = x(k^+)$.

Transitions are triggered by external *events* and *guards*. We denote the finite set of events Σ. It is convenient to partition the events into two types; those that are controllable (they can be assigned a value by the controller), and those that are not. The latter are exogenous and occur spontaneously. Such an event might be specified by nature like a component failure, or a higher level operator who decides to change an operational mode. We will use the symbols s to represent controllable events and p to represent uncontrollable events. Thus, $\Sigma = S \times P$ where $s \in S$ and $p \in P$. An example is given below in Figure 1. A guard is a subset of the continuous state space X that enables a transition. A transition enabled by a guard might represent a protection device. Not all transitions have guards and some transitions might require simultaneous satisfaction of a guard and the occurrence of an event. The guard assignment function is $G : E \to 2^X$.

We consider each discrete state label, $q \in Q$, and each event, $\sigma \in \Sigma$, to be logical variables that take the values True or False. Guards also are specified as logical conditions. In this way the transition system can be defined by a logical specification (formula) $\mathcal{L}$.

In summary, a hybrid control system is composed of:

1) Q, discrete space,
2) X, continuous state space,
3) E, set of transitions,
4) Σ, event set,
5) G, guard assignment function,
6) $\mathcal{L}$, logical specification,
7) F, family of controlled vector fields.

Example 2.1 (Three mode system.): Consider the simple three mode hybrid system shown in Figure 1. Each mode, q_1, q_2, q_3, is characterized by continuous dynamics $x_{k+1} = f_{q_i}(x_k, u_k), \quad i = 1, 2, 3.$

Discrete transitions are associated with the events represented by logical variables p, s_1, s_2, s_3, i.e, $\Sigma = \{p, s_1, s_2, s_3\}$. For example, if the system is in mode q_1 and s_1 evaluates to True, then a mode transition occurs in which the mode changes from q_1 to q_2. In this example, we use two different symbols s and p to denote transition variables to underscore the fact that some transitions are controllable and others not so.

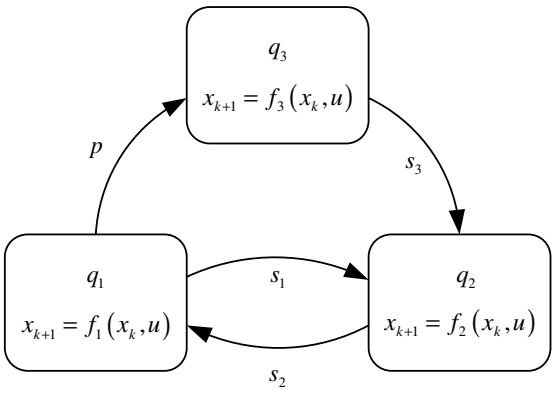

Fig. 1. Three mode hybrid system with controllable and uncontrollable events.

In our formulation the transition system behavior is defined

by the logical specification:

$$\begin{aligned}\mathcal{L} = &\, exactly\left(1, \{q_1(t), q_2(t), q_3(t)\}\right) \wedge \\ &exactly\left(1, \{q_1(t^+), q_2(t^+), q_3(t^+)\}\right) \wedge \\ &\left(q_1(t) \wedge s_1 \Rightarrow q_2(t^+)\right) \wedge \left(q_1(t) \wedge p \Rightarrow q_3(t^+)\right) \wedge \\ &\left(q_1(t) \wedge \neg(s_1 \vee p) \Rightarrow q_1(t^+)\right) \wedge \\ &\left(q_2(t) \wedge s_2 \Rightarrow q_1(t^+)\right) \wedge \left(q_2(t) \wedge \neg s_2 \Rightarrow q_2(t^+)\right) \wedge \\ &\left(q_3(t) \wedge s_3 \Rightarrow q_2(t^+)\right) \wedge \left(q_3(t) \wedge \neg s_3 \Rightarrow q_3(t^+)\right)\end{aligned} \tag{2}$$

Let us dissect this specification. The first and second lines express the fact that the system can only be in one discrete state before the transition (at time t) and after the transition (at time t^+). The next two lines describes all possible transitions from state q_1. Similarly, the last two lines characterize all possible transitions from states q_2 and q_3, respectively.

For computational purposes it is useful to associate with each logical variable, say α, a Boolean variable or indicator function, δ_α, such that δ_α assumes the values 1 or 0 corresponding respectively to α being True or False. It is convenient to define the discrete state vector $\delta_q = [\delta_{q_1}, \ldots, \delta_{q_m}]$, the control event vector $\delta_s = \left[\delta_{s_1}, \ldots, \delta_{s_{m_S}}\right]$, and the exogenous event vector $\delta_p = \left[\delta_{p_1}, \ldots, \delta_{p_{m_P}}\right]$. Precisely one of the elements of δ_q will be unity and all others will be zero.

Notice that with the introduction of the Boolean variables we can replace the set of dynamical equations (1) with the single relation

$$\begin{aligned}x(k+1) &= f(x(k), \delta_q(k), u(k)) \\ &= \delta_{q_1} f_{q_1}(x(k), u(k)) + \cdots \\ &\quad \cdots + \delta_{q_m} f_{q_m}(x(k), u(k))\end{aligned} \tag{3}$$

B. The Control problem

We assume that the system is observed in operation over some finite time horizon T that is divided into N discrete time intervals of equal length. A control policy is a sequence of functions

$$\pi = \left\{\mu_0(x_0, \delta_{q0}), \mu_1(x_1, \delta_{q1}), \ldots, \mu_{N-1}\left(x_{N-1}, \delta_{q(N-1)}\right)\right\}$$

such that

$$[u_k, \delta_{sk}] = \mu_k(x_k, \delta_{qk})$$

Thus, μ_k generates the continuous control u_k and the discrete control δ_k that are to be applied at time k, based on the state (x_k, δ_{qk}) observed at time k.

Consider the set of m-tuples $\{0,1\}^m$. Let Δ_m denote the subset of elements $\delta \in \{0,1\}^m$ that satisfy $\delta_1 + \cdots + \delta_m = 1$. Denote by Π the set of sequences of functions $\mu_k : X \times \Delta_m \to U \times \{0,1\}^{m_S}$ that are piecewise continuous on X.

Now, given the initial state (x_0, δ_{q0}) the problem is to find a policy, $\pi^* \in \Pi$, that minimizes the cost functional

$$J_\pi(x_0, \delta_{q0}) = g_N(x_N, \delta_{qN}) + \sum_{k=0}^{N-1} g_k(x_k, \delta_{qk}, \mu_k(x_k, \delta_{qk})) \tag{4}$$

Definition 2.2 (Optimal Feedback Control Problem): For each $x_0 \in X, \delta_{q0} \in \Delta_m$ determine the control policy $\pi^* \in \Pi$ that minimizes the cost (4) subject to the constraints (1) and (2), i.e.,

$$J_{\pi^*}(x_0, \delta_{q0}) \leq J_\pi(x_0, \delta_{q0}) \quad \forall \pi \in \Pi$$

Notice that if a receding horizon optimal control is desired, once the optimal policy is determined, we need only implement the state feedback control

$$[u_k, \delta_{sk}] = \mu_0(x_k, \delta_{qk}) \tag{5}$$

III. Logical Specification to IP Formulas

Before proceeding to the solution of the optimal control problem we transform the logical specification $\mathcal{L}$ into a set of inequalities involving integer (in fact, Boolean) variables and possibly real variables, so-called *IP-formulas*. The idea of formulating optimization problems using logical constraints and then converting them to IP formulas has a long history [10]. The formulation of complex decision problems often involves specifications and constraints that are most easily stated in terms of logical statements. McKinnon and Williams in [15] proposed an approach that allowed the inclusion of such constraints in conventional optimization methods. They suggested a sequence of transformations that brings a logical specification into a set of IP-formulas. This approach has been refined and generalized in recent years. In [9], [16] the authors present a systematic algorithm for transforming logic formulas into IP formulas. Moreover, they implement their algorithm in *Mathematica*. We have modified and extended these methods in order to obtain simpler and more compact IP formulas.

This concept was more recently used as a means to incorporate qualitative information in process control and monitoring [17], and was more generally introduced into the study of hybrid systems in [8]. Both of these investigations incorporate the method within a model predictive control framework.

For systems of even modest complexity the number of inequalities required can be quite large, so that automation of this process is essential. The basic function is `GenIP` which takes as two arguments, the specification and a list of variables, either propositional variables or bounded real or integer variables. The latter are specified in the form $a \leq x \leq b$. `GenIP` performs a series of transformations and simplifications and returns the IP formulas. A typical usage would look like:

`GenIP`[(q1 $\oplus$ q2) $\wedge$ (qq1 $\oplus$ qq2)$\wedge$((q1 $\wedge$ ($x > 0$)) $\Rightarrow$ qq2)$\wedge$
((q2 $\wedge$ s) $\Rightarrow$ qq1),{q1, q2, qq1, qq2, s,$-2 \leq x \leq 2$}]

$\{1 - \delta_{q1} - \delta_{q2} \geq 0, -1 + \delta_{q1} + \delta_{q2} \geq 0, 1 - \delta_{qq1} - \delta_{qq2} \geq 0,$
$d7 - \delta_{q1} + \delta_{qq2} \geq 0, -1 + \delta_{qq1} + \delta_{qq2} \geq 0,$
$1 - \delta_{q2} + \delta_{qq1} - \delta_s \geq 0, -2 + 2d7 + x \leq 0, -2 \leq x \leq 2,$
$0 \leq d7 \leq 1, 0 \leq \delta_{q1} \leq 1, 0 \leq \delta_{q2} \leq 1, 0 \leq \delta_{qq1} \leq 1,$
$0 \leq \delta_{qq2} \leq 1, 0 \leq \delta_s \leq 1\}$

Notice that propositional variables are replaced by Boolean indicator functions, e.g., q_1 is replaced by δ_{q_1} and new auxiliary variables may be introduced, in this case d7.

Example 3.3 (Three Mode System, Revisited): Consider the logical specification (2). It converts to the set of

IP-formulas:

$$\begin{gathered}
1-\delta_{q_1}-\delta_{q_2}-\delta_{q_3}\geq 0,\quad 1-\delta_{q_1}-\delta_{q_2}-\delta_{q_3}\leq 0\\
1-\delta_{q_1^+}-\delta_{q_2^+}-\delta_{q_3^+}\geq 0,\quad 1-\delta_{q_1^+}-\delta_{q_2^+}-\delta_{q_3^+}\leq 0\\
1-\delta_p-\delta_{q_1}+\delta_{q_3^+}\geq 0\\
1-\delta_{q_1}+\delta_{q_2^+}-\delta_{s_1}\geq 0\\
\delta_p-\delta_{q_1}+\delta_{q_1^+}+\delta_{s_1}\geq 0\\
1-\delta_{q_2}+\delta_{q_1^+}+\delta_{s_1}\geq 0\\
-\delta_{q_2}+\delta_{q_2^+}+\delta_{s_2}\geq 0\\
1-\delta_{q_3}+\delta_{q_2^+}+\delta_{s_3}\geq 0\\
-\delta_{q_3}+\delta_{q_3^+}+\delta_{s_3}\geq 0\\
0\leq\delta_p\leq 1,\quad 0\leq\delta_{q_1}\leq 1,\quad 0\leq\delta_{q_2}\leq 1,\quad 0\leq\delta_{q_3}\leq 1\\
0\leq\delta_{q_1^+}\leq 1,\quad 0\leq\delta_{q_2^+}\leq 1,\quad 0\leq\delta_{q_3^+}\leq 1\\
0\leq\delta_{s_1}\leq 1,\quad 0\leq\delta_{s_2}\leq 1,\quad 0\leq\delta_{s_3}\leq 1
\end{gathered}\tag{6}$$

It is relatively easy to interpret this list. The two inequalities on the first line express the fact the system is in a single discrete state prior to the transition and the second row expresses a similar condition after the transition. The last three rows simply declare that the integer variables can only take the values 0 or 1.

In the specification (2), the transition event p is treated as an exogenous event. In other words, the trigger for transition is completely external to the system. Suppose, however, that the event is associated with a guard, so that the trigger is the entry of the state into a specified region of the continuous state space. To illustrate how this works, suppose that one of the state variables is a bounded real variable $-2\leq x_1\leq 2$. Let the guard be the condition $x_1>1$. The new system specification simply requires that we replace the logical variable p by $(x_1>1)$. Thus, (2) is replaced by

$$\begin{aligned}
\mathcal{L}=\ & exactly\left(1,\left\{q_1\left(t\right),q_2\left(t\right),q_3\left(t\right)\right\}\right)\wedge\\
& exactly\left(1,\left\{q_1\left(t^+\right),q_2\left(t^+\right),q_3\left(t^+\right)\right\}\right)\wedge\\
& \left(q_1\left(t\right)\wedge s_1\Rightarrow q_2\left(t^+\right)\right)\wedge\left(q_1\left(t\right)\wedge\left(x>1\right)\Rightarrow q_3\left(t^+\right)\right)\wedge\\
& \left(q_1\left(t\right)\wedge\neg\left(s_1\vee\left(x>1\right)\right)\Rightarrow q_1\left(t^+\right)\right)\wedge\\
& \left(q_2\left(t\right)\wedge s_2\Rightarrow q_1\left(t^+\right)\right)\wedge\left(q_2\left(t\right)\wedge\neg s_2\Rightarrow q_2\left(t^+\right)\right)\wedge\\
& \left(q_3\left(t\right)\wedge s_3\Rightarrow q_2\left(t^+\right)\right)\wedge\left(q_3\left(t\right)\wedge\neg s_3\Rightarrow q_3\left(t^+\right)\right)
\end{aligned}\tag{7}$$

Conversion to IP-formulas leads to (6) without the three inequalities involving δ_p plus the following additional inequalities:

$$\begin{gathered}
3-4d_4+x>0\\
d_3-\delta_{q_1}+\delta_{q_3^+}\geq 0\\
d_4-\delta_{q_1}+\delta_{q_1^+}+\delta_{s_1}\geq 0\\
-2+d_3+x\leq 0\\
-2\leq x\leq 2,\quad 0\leq d_3\leq 1,\quad 0\leq d_4\leq 1
\end{gathered}\tag{8}$$

Notice that in this reduction two new auxiliary Boolean variables d_3,d_4 are introduced and the real variable x_1 also appears in the formulas.

If all of the guards are linear (set boundaries are composed of linear segments), then the IP formulas are system of linear constraints involving the Boolean variables $\delta_q,\delta_{q^+},\delta_s,\delta_p$, respectively, the discrete state before transition, the discrete state after transition, the controllable events, the exogenous events. They also involve a set of auxiliary Boolean variables, d, introduced during the transformation process, and the continuous state variables, x. The general form is

$$E_5\delta_{q^+}+E_6d\leq E_0+E_1x+E_2\delta_q+E_3\delta_s+E_4\delta_p\tag{9}$$

where the matrices have appropriate dimensions. As we will see in examples below, with $x,\delta_q,\delta_s,\delta_p$ given these inequalities typically provide a unique solution for the unknowns δ_{q^+} and d. The system evolution is described by the closed system of equations (9) and (3).

If the functions f_{q_i} appearing in (3) are all linear, it may be useful to follow the suggestion in [8] and replace (3) by the following simple linear equation

$$x_{k+1}=z_k\tag{10}$$

Where z_k is an auxiliary vector of real variables defined by a conjunction of the logical statements of the form $q_i\Rightarrow z=f_{q_i}\left(x,u\right)$. If this is done, then (9) is replace by

$$E_5\delta_{q^+}+E_6d+E_7z\leq E_0+E_1x+E_2\delta_q+E_3\delta_s+E_4\delta_p\tag{11}$$

In this case, the system is described by (10) and (11).

IV. Constructing the Optimal Solution

The optimal cost is

$$J^*\left(x_0,\delta_{q0}\right)=\min_{\pi\in\Pi}J_\pi\left(x_0,\delta_{q0}\right)$$

and the optimal policy π^* is one that satisfies

$$J_{\pi^*}\left(x_0,\delta_{q0}\right)\leq J_\pi\left(x_0,\delta_{q0}\right)\quad\forall\pi\in\Pi$$

Now we are in a position to apply Bellman's principle of optimality:

Principle of optimality: Suppose $\pi^*=\left\{\mu_1^*,\ldots,\mu_{N-1}^*\right\}$ is an optimal control policy. Then the sub-policy $\pi_i^*=\left\{\mu_i^*,\ldots,\mu_{N-1}^*\right\}$, $1\leq i\leq N-1$ is optimal with respect to the cost function

$$J_\pi\left(x_i,\delta_{qi}\right)=g_N\left(x_N,\delta_{qN}\right)+\sum\nolimits_{k=i}^{N-1}g_k\left(x_k,\delta_{qk},\mu_k\left(x_k,\delta_{qk}\right)\right)$$

Let us denote the optimal cost of the trajectory beginning at x_i,δ_{qi} as $J_i^*\left(x_i,\delta_{qi}\right)$. It follows from the principle of optimality that

$$\begin{aligned}
&J_{i-1}^*\left(x_{i-1},\delta_{q(i-1)}\right)=\\
&\quad\min_{\mu_{i-1}}\left\{g_{i-1}\left(x_{i-1},\delta_{q(i-1)},\mu_{i-1}\right)+J_i^*\left(x_i,\delta_{qi}\right)\right\}
\end{aligned}\tag{12}$$

Equation (12) provides a mechanism for backward recursive solution of the optimization problem. To begin the backward recursion, we need to solve the single stage problem with $i=N$:

$$\begin{aligned}
&J_{N-1}^*\left(x_{N-1},\delta_{q(N-1)}\right)=\\
&\quad\min_{\mu_{N-1}}\left\{g_{N-1}\left(x_{N-1},\delta_{q(N-1)},\mu_{N-1}\right)+J_N^*\left(x_N,\delta_{qN}\right)\right\}
\end{aligned}\tag{13}$$

The end point x_N,δ_{qN} is free, so we begin at a general terminal point

$$\begin{aligned}
&J_{N-1}^*\left(x_{N-1},\delta_{q(N-1)}\right)=\\
&\quad\min_{\mu_{N-1}}\left\{g_{N-1}\left(x_{N-1},\delta_{q(N-1)},\mu_{N-1}\right)+g_N\left(x_N,\delta_{qN}\right)\right\}=\\
&\quad\min_{\mu_{N-1}}\left\{g_{N-1}\left(x_{N-1},\delta_{q(N-1)},\mu_{N-1}\right)+g_N\left(f_{N-1},\delta_{q^+(N-1)}\right)\right\}
\end{aligned}\tag{14}$$

Once the pair μ^*_{N-1}, J^*_{N-1} is obtained, we compute μ^*_{N-2}, J^*_{N-2} from

$$J^*_{N-2}\left(x_{N-2}, \delta_{q(N-2)}\right) = \min_{\mu_{N-2}} \left\{ g_{N-2}\left(x_{N-2}, \delta_{N-2}, \mu_{N-2}\right) + J^*_{N-1}\left(f_{N-2}, \delta_{q^+(N-2)}\right) \right\} \tag{15}$$

Continuing in this way we obtain

$$J^*_{N-i}\left(x_{N-i}, \delta_{q(N-i)}\right) = \min_{\mu_{N-i}} \left\{ \begin{array}{l} g_{N-i}\left(x_{N-i}, \delta_{q(N-i)}, \mu_{N-i}\right) \\ +J^*_{N-i+1}\left(f_{N-i}, \delta_{q^+(N-i)}\right) \end{array} \right\} \tag{16}$$

for $2 \leq i \leq N$.

The process now is straightforward. We need to solve (16) recursively backward, for $i = 2, \ldots, N$ after initializing with (13). We begin by constructing a discrete grid on the continuous state space. The discrete space is denoted $\bar{X}$. At each iteration, working backwards, the optimal control and the optimal cost are evaluated discrete points in $Q \times \bar{X}$. To continue with the next stage we need to set up an interpolation function to cover all points in $Q \times X$.

In order to structure an efficient optimization process we exploit the fact that the system is highly constrained and almost all of the constraints are linear in Boolean variables. The basic approach is as follows:

1) Identify the binary and real variables and separate the inequalities into binary and real sets, binary equations contain only binary variables, real equations can contain both binary and real variables.
2) Use the Mathematica function Reduce to obtain all feasible solutions of the binary inequalities; a list of possible solutions of pairs $\left(\delta_{q^+}, d\right)$. Reduce is a very efficient solver, especially when the inequalities are linear although it is not limited to linear inequalities. In general, if there are N binary variables then there are 2^N combinations that need to be evaluated if one were to attempt to optimize by enumeration. But the feasible combinations are almost certainly much fewer. In the simple example below, there are 8 variables or 256 combinations, but only 8 are feasible. Reduce identifies these very rapidly.
3) Use Reduce to solve the real inequalities for the real variables for every feasible combination of binary variables. Many of these combinations of binary variables will not admit feasible real variables, so they can be dropped. The remaining combinations typically produce unique values for the real variables.
4) Enumerate the values of the cost for each feasible pair of binary and real variables and select the minimum.

V. Applications

Power electronic devices are ideal candidates for the application of the new concepts and tools of hybrid control theory. We will consider two different applications. First, a power conditioning system of a type often used to isolate devices, like motors that have large, short period power requirements, from a primary power source that has limited power supply capability. It is a simple, transparent and useful application. Somewhat more complex is a DC-DC converter. This device has attracted the interest of many investigators, including the recent work [18], [19], [20], [21], and can provide a comparative look at different control formulations. Finally, we consider a simple power management system, emphasizing the approach to problem formulation.

A. Power Conditioning Systems

A power conditioning system is shown in Figure 2. Its purpose is to insure that the current demand on the DC source is limited even though the load current may be quite large for short periods of time. The problem is the design of the switching strategy.

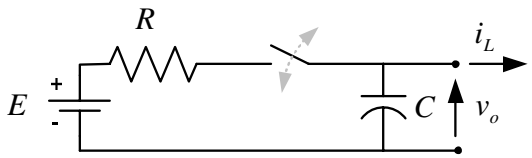

Fig. 2. A simple power conditioning system.

1) Modeling: A hybrid automaton model of the system without a specified control strategy is shown in Figure 3 where q is the capacitor charge. In this open loop configuration, the events are not enabled by a guard, but by an externally generated event - the switch. The proposition s denotes 'the switch is closed'.

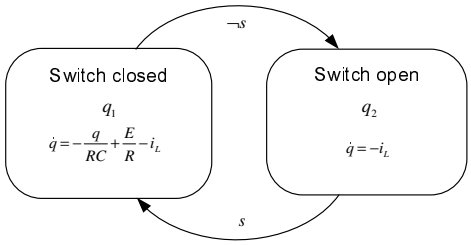

Fig. 3. Hybrid automaton for the power conditioning system.

The specification for the hybrid automaton in Figure 3 is

$$\begin{array}{l} \mathcal{L} = (q_1 \oplus q_2) \wedge \left(q_1^+ \oplus q_2^+\right) \wedge \\ \quad \left(q_1 \wedge \neg s \Rightarrow q_2^+\right) \wedge \left(q_1 \wedge s \Rightarrow q_1^+\right) \wedge \\ \quad \left(q_2 \wedge s \Rightarrow q_1^+\right) \wedge \left(q_2 \wedge \neg s \Rightarrow q_2^+\right) \end{array} \tag{17}$$

The corresponding IP formulas are:

$$\begin{array}{c} 1 - \delta_{q_1^+} - \delta_{q_2^+} \geq 0 \\ -1 + \delta_{q_1^+} + \delta_{q_2^+} \geq 0 \\ 1 - \delta_{q_1} - \delta_{q_2} \geq 0 \\ -1 + \delta_{q_1} + \delta_{q_2} \geq 0 \\ 1 - \delta_s - \delta_{q_1} + \delta_{q_2^+} \geq 0 \\ 1 - \delta_s - \delta_{q_2} + \delta_{q_2^+} \geq 0 \\ \delta_s - \delta_{q_1} + \delta_{q_1^+} \geq 0 \\ \delta_s - \delta_{q_2} + \delta_{q_1^+} \geq 0 \\ 0 \leq \delta_s \leq 1, 0 \leq \delta_{q_1} \leq 1, 0 \leq \delta_{q_2} \leq 1 \\ 0 \leq \delta_{q_1^+} \leq 1, 0 \leq \delta_{q_2^+} \leq 1 \end{array} \tag{18}$$

In discrete time form, the dynamics of the system can be written as

$$\begin{aligned} q_{k+1} = \quad & \delta_{q_1}\left(e^{-\frac{\Delta t}{RC}} q_k + \left[1 - e^{-\frac{\Delta t}{RC}}\right](CE - RCi_{L,k})\right) \\ & + \delta_{q_2}\left(q_k - i_{L,k}\Delta t\right) \end{aligned} \tag{19}$$

For specificity in later calculations take $E = 1, C = 1, R = 1$, and $\Delta t = 0.05$

$$q_{k+1} = \delta_1\left(0.9512\, q_k + .0488\,(1 - i_L)\right) + \delta_2\left(q_k - 0.05\, i_L\right) \tag{20}$$

We can simplify this equation by introducing a new variable z defined by the specification

$$\begin{aligned} & \left(q_1^+ \Rightarrow z = 0.9512\, q_k + .0488\,(1 - i_L)\right) \\ & \wedge \left(q_2^+ \Rightarrow z = q_k - 0.05\, i_L\right) \end{aligned} \tag{21}$$

This allows us to write

$$q_{k+1} = z \tag{22}$$

The current i drawn from the source is defined by the specification

$$\left(q_1^+ \Rightarrow i = -q + 1\right) \wedge \left(q_2^+ \Rightarrow i = 0\right) \tag{23}$$

By adjoining (22) and (23) to (17) with bounds on the real variables i, q, z

$$-1 \le i \le 1, 0 \le q \le 2, -1 \le z \le 3$$

Then, in addition to (18), we obtain

$$\begin{gathered} d_3 - \delta_{q_1^+} \ge 0 \\ d_4 - \delta_{q_1^+} \ge 0 \\ -1 + d_2 + \delta_{q_1^+} \ge 0 \\ 1 - d_1 + i \ge 0 \\ 1 - 2d_3 + i + q \ge 0 \\ 3 - 3d_2 - q + z \ge 0 \\ 2.9024 - 2.9512d_4 - 0.95512q + z \ge 0 \\ -1 + d_1 + i \le 0 \\ -3 + 2d_3 + i + q \le 0 \\ -3 + 3d_2 - q - z \le 0 \\ -3 + 2.9512d_4 - 0.9512q + z \le 0 \\ 0 \le d_1 \le 1, 0 \le d_2 \le 1, 0 \le d_3 \le 1, 0 \le d_4 \le 1 \\ -1 \le i \le 1, -1 \le z \le 3, 0 \le q \le 2 \end{gathered} \tag{24}$$

2) Optimal Control: The optimization problem is formulated as follows. We consider the operation of the system over a time period of 0.5 sec (10 time steps, $\Delta t = 0.05$). Generally, the resistance R is very small so it is expected that for reasonable deviations of capacitor bank voltage from the nominal value E, currents from the DC supply will be large when switch is closed. The goal of the controller is to open and close the switch to achieve capacitor resupply, while insuring a reasonable average current (about 1 amp, in this case) over the specified time period.

To accomplish this we specify a cost function

$$J = \alpha\left[q_N - \bar{q}\right]^2 + \frac{1}{N}\sum_{k=0}^{N-1}\left(i_k(k+1)\right)^{10} \tag{25}$$

with $N = 10$. Notice that the cost trades a terminal cost that penalizes any deviation of capacitor bank charge (equivalently, voltage) from its nominal value against an accumulated charge current cost. The current cost is subjected to a time dependent weighting. The weighting function adds flexibility and is useful in this example. Using the weighting function guarantees the switch is closed for some time (however short), while insuring a limited average current. Using the 10^{th} power rather than a more common quadratic implies, that currents less than $1/(k+1)$ are penalized very little, while currents greater than $1/(k+1)$ are very costly.

The optimal control is obtained by minimizing the cost J in (25) subject to the dynamical constraints (22) and the inequality constraints (18), (24). The result is a discrete controller in which the switch is open or closed depending on the value of capacitor charge and time on the interval $t \in [0, 1]$. The control is to be applied periodically. Figure (4) shows a

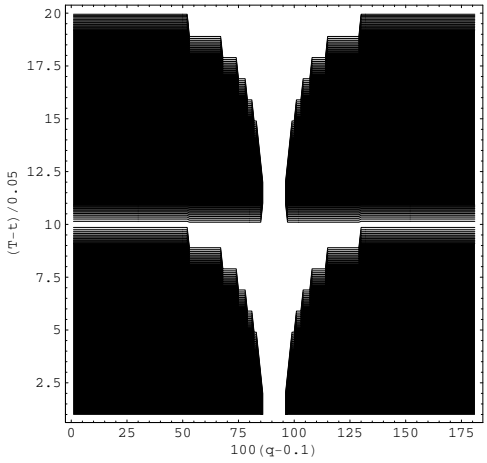

Fig. 4. Two cycles of a periodic switching controller. The switch is open in the black regions, closed in the white.

typical result (corresponding to the case of zero load current, $i_L = 0$). The black region corresponds to an open switch, and the white to a closed switch. Two cycles of the periodic control is shown.

Note that the result is similar to a pulse width modulated control in which the duty cycle varies with the capacitor charge.

B. DC-DC Converter

The DC-DC boost converter to be considered is shown in Figure 5. There are four possible switch-diode arrangements: switch open and diode conducting, switch open and diode nonconducting, switch closed and diode nonconducting, switch closed and diode conducting. It is common to assume that $q \ge 0$. This is justified by the fact that the only way it is is possible to have $q < 0$ would be to initialize the capacitor this way. If this assumption is made, then the last operating condition can be discarded. Another common assumption is that the inductor current is always positive $i > 0$, which would eliminate the third arrangement. However, it is certainly possible to have $i = 0$ for a nontrivial time period. For example, suppose the switch is opened after the capacitor voltage reaches a value significantly above E. Then the inductor current will begin decreasing and could reach $i = 0$ before the capacitor

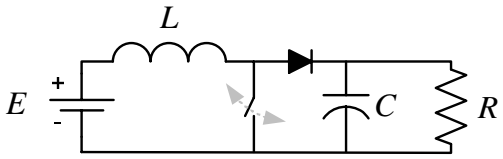

Fig. 5. The hybrid automaton model of the DC-DC converter.

current drops below E, at which time the diode would become nonconducting while the voltage continues to drop. Thus, we consider the three mode model shown in Figure 6, as in [18]. The logical specification is

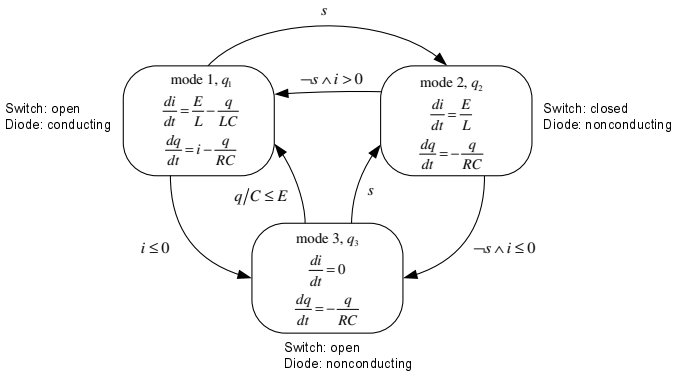

Fig. 6.

$$
\begin{aligned}
\mathcal{L} = & \, exactly\,(1,\{q_1,q_2,q_3\}) \wedge \\
& exactly\left(1,\left\{q_1^+,q_2^+,q_3^+\right\}\right) \wedge \\
& \left(q_1 \wedge s \Rightarrow q_2^+\right) \wedge \left(q_1 \wedge (i \le 0) \Rightarrow q_3^+\right) \wedge \\
& \left(q_1 \wedge \neg(s \vee (i \le 0)) \Rightarrow q_1^+\right) \wedge \\
& \left(q_2 \wedge (\neg s \wedge (i > 0)) \Rightarrow q_1^+\right) \wedge \\
& \left(q_2 \wedge (\neg s \wedge (i \le 0)) \Rightarrow q_3^+\right) \wedge \\
& \left(q_2 \wedge \neg((\neg s \wedge (i \le 0)) \vee (\neg s \wedge (i > 0))) \Rightarrow q_2^+\right) \wedge \\
& \left(q_3 \wedge s \Rightarrow q_2^+\right) \wedge \left(q_3 \wedge (q \le 1) \Rightarrow q_1^+\right) \wedge \\
& \left(q_3 \wedge \neg(s \vee (q \le 1)) \Rightarrow q_3^+\right)
\end{aligned}
$$

From which we obtain the IP formulas

$$
\begin{gathered}
1-\delta_{q_1}-\delta_{q_2}-\delta_{q_3} \ge 0, \quad 1-\delta_{q_1}-\delta_{q_2}-\delta_{q_3} \le 0 \\
1-\delta_{q_1^+}-\delta_{q_2^+}-\delta_{q_3^+} \ge 0, \quad 1-\delta_{q_1^+}-\delta_{q_2^+}-\delta_{q_3^+} \le 0 \\
1-\delta_{q_1}-\delta_{q_2^+}-\delta_s \ge 0, \quad 1-\delta_{q_3}-\delta_{q_2^+}-\delta_s \ge 0 \\
d_1-\delta_{q_1}+\delta_{q_3^+} \ge 0, \quad d_5-\delta_{q_3}+\delta_{q_1^+} \ge 0, \\
d_3+d_4-\delta_{q_2}+\delta_{q_2^+} \ge 0 \\
d_2-\delta_{q_1}+\delta_{q_1^+}+\delta_s \ge 0, \quad d_7-\delta_{q_2}+\delta_{q_1^+}+\delta_s \ge 0 \\
d_8-\delta_{q_2}+\delta_{q_2^+}+\delta_s \ge 0, \quad d_6-\delta_{q_3}+\delta_{q_3^+}+\delta_s \ge 0 \\
0 \le d_1,d_2,d_3,d_4,d_5,d_6,d_7,d_7 \le 1 \\
0 \le \delta_{q_1},\delta_{q_1^+},\delta_{q_2},\delta_{q_2^+},\delta_{q_3},\delta_{q_3^+},\delta_s \le 1 \\
2-2d_1+i>0, \quad 2-2d_4+i>0 \\
-1+d_2+i \le 0, \quad -1+d_3+i \le 0, \\
-1+d_7+i \le 0, \quad -1+d_8+i \le 0 \\
1-2d_5+q>0, \quad -2+d_6+q \le 0 \\
-1 \le i \le 1, \quad 0 \le q \le 1
\end{gathered}
$$

1) Power Management System: We briefly describe a problem of current interest to us. Figure 7 constitutes a benchmark derived from the notional DD(X) power system configuration. Consider a scenario in which various faults can afflict the system; in which case the power management system should reconfigure the system to maintain a maximum level of functionality. As a simple illustrative example, suppose a fault occurs that results in the isolation of bus 5, so that the post fault systems reduces to that shown in Figure 8. Now a second fault occurs that removes half of the transmission capacity between busses 2 and 3. All numbers are 'per unit' with the following base values:

$$
\begin{aligned}
V_{base} &= \tfrac{13.8}{\sqrt{3}}\,kV = 7.97\,kV \\
P_{base} &= \tfrac{36}{3}\,MVA = 12\,MVA \\
I_{base} &= \tfrac{P_{base}}{V_{base}} = \tfrac{12\sqrt{3}}{13.8}A = 1.51A \\
Z_{base} &= \tfrac{(V_{base})^2}{P_{base}} = \left(\tfrac{13.8}{\sqrt{3}}\right)^2 \tfrac{1}{12}\ \Omega = 5.29\ \Omega
\end{aligned}
$$

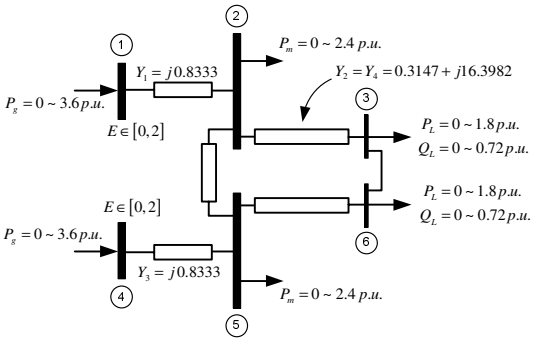

Fig. 7. The benchmark example derived from the notional DD(X) IPS.

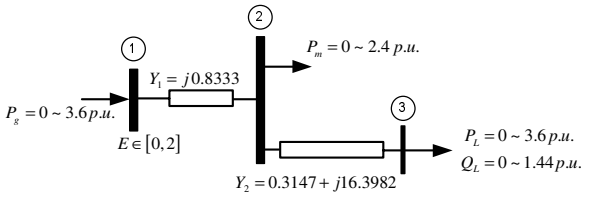

Fig. 8. The system following a fault that results in disconnection of bus 5.

The network model is a classical model generated automatically in Mathematica. Let $\delta_i,\ i = 1,2,3$ denote the angles at bus 1,2,3, respectively. Choose bus 1 as a reference and define $\theta_2 = \delta_2 - \delta_1,\ \theta_3 = \delta_3 - \delta_1$. Then the network equations are

$$
\begin{aligned}
0 = & P_2 - b_{12}V_1V_2\sin\theta_2 - g_{22}V_2^2 - g_{23}V_2V_3\cos(\theta_2-\theta_3) \\
& -b_{23}V_2V_3\sin(\theta_2-\theta_3) \\
0 = & P_3 - g_{33}V_3^2 - g_{23}V_2V_3\cos(\theta_2-\theta_3) - b_{23}V_2V_3\sin(\theta_2-\theta_3) \\
0 = & Q_2 - b_{12}V_1V_2\cos\theta_2 - b_{22}V_2^2 \\
& -b_{23}V_2V_3\cos(\theta_2-\theta_3) - g_{23}V_2V_3\sin(\theta_2-\theta_3) \\
0 = & Q_3 - b_{33}V_3^2 - b_{23}V_2V_3\cos(\theta_2-\theta_3) - g_{23}V_2V_3\sin(\theta_2-\theta_3)
\end{aligned}
$$

In addition, we have

$$
\begin{aligned}
& P_2 = -P_m,\ Q_2 = 0 \\
& P_3 = -P_L,\ Q_3 = -Q_L \\
& V_1 = E
\end{aligned}
$$

The load is actually an aggregate of many different types of loads including motors, lighting and heating. The power consumption depends on the applied voltage. When a disturbance occurs various controllers take action that tends to restore the

power consumption. The following model is used for the load.

$$\begin{aligned}\dot{\sigma} &= -\tfrac{\sigma}{T} - 2v \\ P_L &= (1-\eta_L)\, P_0 \left(1 + \tfrac{\sigma}{T} + 2v\right) \\ Q_L &= (1-\eta_L)\, Q_0 \left(1 + \tfrac{\sigma}{T} + 2v\right)\end{aligned}$$

where η_L is the load shedding fraction. The load shed fraction changes in accordance with the evolution of the hybrid automaton shown in Figure 9.

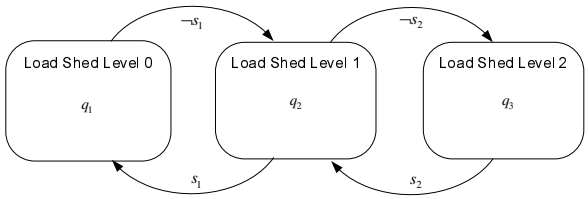

Fig. 9. Diagram showing the load shedding transition logic. There are two sets of circuit breakers. s_1, s_2 are logical statements that the first or second sets, respectively, are closed. From load shed level 0, set 1 can be opened to and the system transitions to load shed level 1 from which the second set of breakers can be opened to drop additional load.

The specification for the hybrid automaton is

$$\begin{aligned}\mathcal{L} = exactly(1, \{q_1(t), q_2(t), q_3(t)\}) \wedge \\ exactly(1, \{q_1(t^+), q_2(t^+), q_3(t^+)\} \wedge \\ (q_1(t) \wedge \neg s_1 \Rightarrow q_2(t^+)) \wedge (q_2(t) \wedge \neg s_2 \Rightarrow q_3(t^+)) \wedge \\ (q_3(t) \wedge s_2 \Rightarrow q_2(t^+)) \wedge (q_2(t) \wedge s_1 \Rightarrow q_1(t^+)) \wedge \\ (q_1(t) \wedge s_1 \Rightarrow q_1(t^+)) \wedge (q_2(t) \wedge s_2 \Rightarrow q_2(t^+)) \wedge \\ (q_2(t) \wedge \neg s_1 \Rightarrow q_2(t^+)) \wedge (q_3(t) \wedge \neg s_2 \Rightarrow q_3(t^+))\end{aligned}$$

The IP formulas are

$$\begin{gathered}1 - \delta_{q_1} - \delta_{q_2} - \delta_{q_3} \geq 0,\ -1 + \delta_{q_1} + \delta_{q_2} + \delta_{q_3} \geq 0 \\ 1 - \delta_{q_1^+} - \delta_{q_2^+} - \delta_{q_3^+} \geq 0,\ -1 + \delta_{q_1^+} + \delta_{q_2^+} + \delta_{q_3^+} \geq 0 \\ 1 - \delta_{q_1} + \delta_{q_1^+} - \delta_{s_1} \geq 0,\ 1 - \delta_{q_2} + \delta_{q_1^+} - \delta_{s_1} \geq 0 \\ 1 - \delta_{q_2} + \delta_{q_2^+} - \delta_{s_2} \geq 0,\ 1 - \delta_{q_3} + \delta_{q_2^+} - \delta_{s_2} \geq 0 \\ -\delta_{q_1} + \delta_{q_2^+} + \delta_{s_1} \geq 0,\ -\delta_{q_2} + \delta_{q_2^+} + \delta_{s_1} \geq 0 \\ -\delta_{q_2} + \delta_{q_3^+} + \delta_{s_2} \geq 0,\ -\delta_{q_3} + \delta_{q_3^+} + \delta_{s_2} \geq 0 \\ 0 \leq \delta_{q_1} \leq 1, 0 \leq \delta_{q_2} \leq 1, 0 \leq \delta_{q_3} \leq 1 \\ 0 \leq \delta_{q_1^+} \leq 1, 0 \leq \delta_{q_2^+} \leq 1, 0 \leq \delta_{q_3^+} \leq 1 \\ 0 \leq \delta_{s_1} \leq 1, 0 \leq \delta_{s_2} \leq 1\end{gathered}$$

VI. Conclusions

We have described an approach to modeling power systems as hybrid dynamical systems that include continuous (ODE or DAE) and discrete (FSA) subsystems. The essential feature of the model is a characterization of the discrete subsystem in terms of a set of IP formulas. The application of this model to the design of optimal feedback control systems using dynamic programming has also been described. Computational tools for performing the translation of the logical specification to IP formulas and for solving a limited form of the dynamic programming problem have been assembled in *Mathematica*. Examples have been given.

Acknowledgment

This research was supported by the Office of Naval Research Contract Number N00014-04-M-0285 and the National Science Foundation Contract Number ECS-0400391.

References

[1] H. G. Kwatny, A. K. Pasrija, and L. Y. Bahar, "Static bifurcations in electric power networks: Loss of steady state stability and voltage collapse," *IEEE Transactions on Circuits and Systems*, vol. CAS33, no. 10, p. 981991, 1986.

[2] I. Dobson and H.-D. Chiang, "Towards a theory of voltage collapse in electric power systems," *Systems and Control Letters*, vol. 13, p. 253262, 1989.

[3] C. A. Caizares and F. L. Alvarado, "Computational experience with the point of collapse method on very large ac/dc systems," in *Bulk Power Systems Voltage II PhenomenaVoltage Stability and Security*, Deep Creek Lake, MD, 1991, pp. 103–112.

[4] V. Venkatasubramanian, H. Schttler, and J. Zaborsky, "A taxonomy of the dynamics of the large power system with emphasis on its voltage stability," in *Bulk Power System Voltage Phenomena II: Voltage Stability and Security*, L. H. Fink, Ed., Deep Creek Lake, MD, 1991, p. 944.

[5] J. E. Hopcroft and J. D. Ullman, *Introduction to Automata Theory, Languages and Computation*. Reading: AddisonWesley, 1979.

[6] T. Geyer, M. Larsson, and M. Morari, "Hybrid emergency voltage control in power systems," in *European Control Conference*, Cambridge, 2003.

[7] P. J. Antsaklis, *Proceedings of the IEEE, Special issue on Hybrid Systems: Theory and Application*, vol. 88, no. 7, 2000.

[8] A. Bemporad and M. Morari, "Control of systems integrating logic, dynamics, and constraints," *Automatica*, vol. 35, no. 3, pp. 407–427, 1999.

[9] Q. Li, Y. Guo, and T. Ida, "Transformation of logical specification into ip-formulas," in *3rd International Mathematica Symposium (IMS '99)*. Hagenburg, Austria: Computational Mechanics Publications, WIT Press, 1999.

[10] H. P. Williams, *Model Building in Mathematical Programming*. John Wiley and Sons, 1993.

[11] M. S. Branicky, V. S. Borkar, and S. K. Mitter, "A unified framework for hybrid control: Model and optimal control theory," *IEEE Transactions on Automatic Control*, vol. 43, no. 1, pp. 31–45, 1998.

[12] H. J. Sussmann, "A maximum principle for hybrid optimal control problems," in *Conference on Decision and Control*, Phoenix, AZ, 1999, pp. 425–430.

[13] J. Lu, L.-Z. Liao, and J. H. Taylor, "Optimal control of systems with continuous and discrete states," in *Conference on Decision and Control*, San Antonio, TX, 1993, pp. 2292–2297.

[14] S. Hedlund and A. Rantzner, "Optimal control of hybrid systems," in *Conference on Decision and Control*, Pheonix, AZ, 1999, pp. 3972–3977.

[15] K. McKinnon and H. Williams, "Constructing integer programming models by the predicate calculus," *Annals of Operations Research*, vol. 21, pp. 227–246, 1989.

[16] Q. Li, Y. Guo, and T. Ida, "Modelling integer programming with logic: Language and implementation," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E83-A, no. 8, pp. 1673–1680, 2000.

[17] M. L. Tyler and M. Morari, "Propositional logic in control and monitoring problems," *Automatica*, vol. 35, no. 4, pp. 565–582, 1999.

[18] P. Gupta and A. Patra, "Hybrid sliding mode control of dc-dc power converters," in *IEEE Tencon 2003*. Bangalore: Allied Publishers, 2003.

[19] M. Senesky, G. Eirea, and T. J. Koo, "Hybrid modeling and control of power electronics," in *Hybrid Systems: Computation and Control*, ser. Lecture Notes in Computer Science. New York: Springer-Verlag, 2003, vol. 2623, pp. 450–465.

[20] B. Lincoln and A. Rantzer, "Relaxing dynamic programming," 2003.

[21] T. Geyer, G. Papafotiou, and M. Morari, "On the optimal control of switch-mode dc-dc converters." in *HSCC*, ser. Lecture Notes in Computer Science, vol. 2993. Springer, 2004, pp. 342–356.